

REMARKS

This application has been carefully considered in connection with the Examiner's Final Office Action dated November, 2007. Reconsideration and allowance are respectfully requested in view of the following.

Summary of Rejections

Claims 1-37 were pending at the time of the Office Action.

Claims 1-10, 12, 13, 15-23, and 31-37 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,159,224 B2 to Sharma et al. (hereinafter "Sharma") in view of U.S. Publication No. 2002/0032783 A1 to Tuatini (hereinafter "Tuatini").

Claims 11 and 14 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini in further view of U.S. Publication No. 2005/0223392 A1 to Cox et al. (hereinafter "Cox").

Claims 24-30 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini as applied to claim 23 above, and further in view of U.S. Publication No. 2006/0036448 A1 to Haynie et al (hereinafter "Haynie").

Summary of Response

Claims 1 and 33-37 remain as previously presented.

Claims 2-32 remain as originally submitted.

Remarks and Arguments are provided below.

Summary of Claims Pending

Claims 1-37 are currently pending following this response.

Improper Final Rejection

Applicants respectfully submit that the finality of the Final Office Action dated November 15, 2007 is improper. MPEP 706.07(a) states, “second or any subsequent actions on the merits shall be final, **except where the examiner introduces a new ground of rejection that is neither necessitated by applicant’s amendment of the claims, nor based on information submitted in an information disclosure statement**” (emphasis added). Applicants respectfully submit that the new ground of rejection introduced by the Examiner against at least independent claim 32 is neither necessitated by Applicants’ amendment of the claims, nor based on information submitted in an information disclosure statement. In the Office Action dated July 3, 2007, claim 32 was rejected under 35 U.S.C. 103(a) by Sharma in view of Cox. In the Final Office Action dated November 15, 2007, claim 32 was rejected under 35 U.S.C. 103(a) by Sharma in view of Tuatini. Applicants note that claim 32 remains as originally filed. Accordingly, Applicants respectfully submit that the finality of the Final Office Action dated November 15, 2007 is premature and respectfully requests the pending application be withdrawn from final rejection.

Response to Rejections under Section 103

According to the United States Supreme Court in *Graham v. John Deere Co. of Kansas City*, an obviousness determination begins with a finding that **“the prior art as a**

whole in one form or another contains all” of the elements of the claimed invention.

See *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 22 (U.S. 1966).

If a proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984). See MPEP 2143.01(V). Also, if the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). See MPEP 2143.01(VI). Further, the prior art can be modified or combined to reject claims as *prima facie* obvious as long as there is a reasonable expectation of success. *In re Merck & Co., Inc.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). See MPEP 2143.02(I).

Claim 1:

In the Office Action dated November 15, 2007, Claim 1 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini.

Independent claim 1 recites:

1. A method of accessing an Enterprise Java Bean (EJB) by a non-Java application within a computing environment, comprising:
 - a) **making a call, by the non-Java application, to a client library**, wherein the call includes input parameters;
 - e) **invoking a function within the client library to construct an HTTP request** corresponding to the input parameters of the call made from the non-Java application;
 - f) passing the HTTP request from the client library to an EjbServlet;
 - g) invoking a method on an EJB by the EjbServlet based upon the HTTP request;

- e) returning information from the invoked method from the EJB to the EjbServlet;
- f) decoding the returned information into an HTTP response string by the EjbServlet;
- g) transmitting the HTTP response from the EjbServlet to the client library; and
- h) **parsing and converting the HTTP response by the client library** into return information compatible with the non-Java application and then passing the return information from the client library to the non-Java application.

Applicants respectfully submit that the art of record does not establish a *prima facie* case of obviousness as to the pending claims because the art of record fails to teach or suggest all of the claim limitations. Specifically, Sharma in view of Tuatini fails to teach or suggest a non-Java application, a client library, or parsing and converting the HTTP response by the client library into return information compatible with the non-Java application.

Further, Applicants respectfully submit that the art of record does not establish a *prima facie* case of obviousness as to the pending claims because the combination of Sharma in view of Tuatini would render Sharma unsatisfactory for its intended purpose, change the principle of operation of Sharma, and would not have a reasonable expectation of success.

I. Sharma in view of Tuatini does not teach or suggest a non-Java application.

The Final Office Action relied on disclosure of the client side proxy or the dynamic proxy of Sharma to read on the claimed application. As noted in the arguments of section I in the response to the Office Action filed on September 21, 2007, the client side proxy or the dynamic proxy of Sharma is a Java application. For clarity, column 8, lines 40-53 of Sharma are reproduced below:

"As previously described, an instance of a stub class may represent a client side proxy or a stub instance for a service endpoint. For a **client side proxy**, client 130 **may utilize a dynamic proxy class** that supports a service endpoint interface dynamically at runtime without requiring code generation of a stub class that implements a specific service endpoint interface. **The creation of a dynamic proxy may be supported by the getPort method defined in the javax.xml.rpc.Service interface.** A serviceEndpointInterface parameter associated with this method may specify the service endpoint interface that is supported by the created dynamic proxy. The dynamic proxy may be used by client 130 to invoke an operation on a target service endpoint defined by server 110."

On page 3, the Final Office Action concedes, "Sharma is silent with reference to a non-Java application." The Final Office Action relied on disclosure in paragraph [0120] of Tuatini to provide teaching of a non-Java application. Tuatini discloses a framework for enabling clients to access functions of shared services through a messaging component (Tuatini: Fig. 39; paragraph [0114] and [0118]). Tuatini further discloses in paragraph [0126] that each shared service may share a common XML format for messages or that different services could use different formats. Tuatini discloses in paragraph [0126], "For example, Java-based shared services could use Java objects as messages, and other shared services could use XML messages." Paragraph [0120] of Tuatini, which was relied on by the Final Office Action, states, "The shared services and clients can also include applications supporting various languages and technologies (e.g., both Enterprise JavaBean (EJB) components and non-Java components)." Based on this disclosure, Tuatini discloses that a system may include an application that supports EJB components and an application that supports non-Java components. Also, based on the above cited disclosure of Tuatini and in consideration of the disclosure of Tuatini as a whole, it would appear that applications that support non-Java components would be applications that support XML. Accordingly, Tuatini does not teach or suggest a non-Java application as

claimed, but rather teaches an application that supports non-Java components. The specification discloses in paragraph 0011, "The non-Java application of the present disclosure can be any computing application implementing a business-related functionality written in a non-Java language that requires access to external functionality encapsulated in an EJB." Paragraph 0011 further provides some examples of non-Java applications as FORTRAN, Visual Basic, C, Pascal, Basic, and ClearBasic. Further, while Tuatini may disclose the existence of an application that supports non-Java components (i.e., XML components), Tuatini does not teach or suggest accessing an EJB by a non-Java application as claimed.

Accordingly, Sharma in view of Tuatini does not teach or suggest a non-Java application.

II. The proposed modification of Sharma would render Sharma unsatisfactory for its intended purpose, change the principle of operation of Sharma, and would not have a reasonable expectation of success.

Assuming arguendo that Sharma in view of Tuatini does disclose a non-Java application, the proposed modification would render Sharma unsatisfactory for its intended purpose, change the principle of operation of Sharma, and would not have a reasonable expectation of success. The Final Office Action proposed to modify Sharma by changing the Java application of Sharma to a non-Java application. Applicants note that Sharma discloses a Java client with a Java runtime system that receives a WSDL from a service endpoint and uses a WSDL-to-Java mapping tool to create a stub class (i.e., object oriented class). The Java client may use the javax.xml.rpc.Service interface to create a client side proxy or dynamic proxy that may be used by the Java client to

invoke an operation on the service endpoint. The Java client creates a call object using the javax.xml.rpc.Service interface and the javax.xml.rpc.Call interface to invoke operations on the service endpoint (Sharma: column 7, line 30 - column 8, line 61). Accordingly, the intended purpose of Sharma is to utilize a WSDL in a Java client and runtime system to access a service endpoint (Sharma: Abstract). How can Sharma accomplish the intended purpose with a non-Java application? How can a non-Java application replace the Java client and runtime system of Sharma? Further, Applicants respectfully submit that such a modification would drastically change the principle of operation of Sharma and would not have a reasonable expectation of success. A non-Java application would not be able to perform any of the functionality or use any of the Java interfaces disclosed by Sharma.

Accordingly, the proposed modification of Sharma would render Sharma unsatisfactory for its intended purpose, change the principle of operation of Sharma, and would not have a reasonable expectation of success.

III. Sharma in view of Tuatini does not teach or suggest a client library.

The Final Office Action does not appear to directly indicate which element(s) of Sharma are being read on the claimed client library. Based on the Final Office Action and the disclosure of Sharma, it appears that the client 130 or the client side runtime system 134 of Sharma is being interpreted as the claimed (non-java) application and the client side proxy or dynamic proxy is being interpreted as the claimed client library. Applicants note column 8, lines 51-53 of Sharma disclose, "The dynamic proxy may be used by client 130 to invoke an operation on a target service endpoint defined by server 110."

Applicants respectfully submit that the client side proxy or dynamic proxy is not a client library. Further, Sharma does not provide any teaching or suggestion of a client library.

Applicants respectfully refer to paragraphs 0013-0019 of the specification for a discussion of the claimed client library. Paragraph 0014 of the specification discloses, "In an embodiment, the client library 30 is a linkable library that exposes an API ... to allow the applications to make method calls on an EJB." Paragraph 0014 of the specification further discloses, "The client library 30 may be implemented as a single shared object library file" Paragraph 0015 of the specification discloses, "In an embodiment, [t]he client library 30 is ... dynamically linked to that application." Accordingly the client library is a module of code that is linked or otherwise interfaced with the non-Java application. For example, in a Unix environment, the client library may be a .so file. Similarly, in a windows environment, the client library may be a .dll file. One skilled in the art of computer programming at the time of the invention would clearly recognize that the client proxy or dynamic proxy of Sharma is not a client library as claimed. Applicants respectfully submit that Tuatini similarly does not disclose a client library.

Accordingly, Sharma in view of Tuatini does not teach or suggest a client library.

IV. Sharma in view of Tuatini does not teach or suggest parsing and converting the HTTP response by the client library into return information compatible with the non-Java application.

The Final Office Action states on page 3, "Sharma is silent with reference to ... parsing and converting the HTTP response by the client library into return information compatible with the non-Java application and then passing the return information from the client library to the non-Java application."

The Final Office Action relied on paragraphs 0119, 0127, and 0138 of Tuatini to read on these limitations. Paragraph 0019 of Tuatini discloses that an external client may access shared services by providing messages through a passthru component to a messaging component (see Fig. 39 of Tuatini). Tuatini discloses that the passthru component may process the messages to a format appropriate for the messaging component. The passthru component may also return responses to the requesting client in the appropriate manner. Claim 1 does not generically recite returning the HTTP response to the non-Java application in an appropriate manner, but rather specifically recites that the client library **parses and converts** the HTTP response. Paragraph 0019 of Tuatini does not provide any teaching or suggestion of parsing or converting as claimed. Further, Applicants respectfully submit that the passthru component is not a client library as claimed.

Paragraph 0127 of Tuatini discloses providing XML document type definition (DTD) documents for each access interface function of a shared service. Paragraph 0127 further discloses providing a DTD for response messages that are returned to a requesting client. Applicants respectfully submit that disclosure of a DTD that describes an XML response generated by a shared service is not disclosure of parsing and converting the HTTP response as claimed. While a DTD may be used to properly interpret tags in an XML document, a DTD itself does not parse an XML document and it does not perform conversions on data.

Paragraph 0138 of Tuatini discloses, "For each response message received, the transport connector performs the same processing discussed (e.g., translation or additional security measures) in order to send the response message back to the

requesting client.” Applicants respectfully note paragraph 0136 of Tuatini discloses, “[T]he transport connector ... then determines the form of the sub-message (e.g., JavaBean or serialized XML) received and determines if the function of the shared service can accept that form. If not, the transport connector invokes the translator to translate the sub-message as necessary.” Applicants note that the transport connector is part of the messaging component discussed above and depicted in Fig. 39 of Tuatini. Accordingly this disclosed translation is occurring between the messaging component and the shared services and not at the client library as claimed. Further, Tuatini only discloses translating between, for example, a JavaBean object to an XML format or vice versa, or between an XML format to SQL calls. Tuatini does not provide any teaching or suggestion of parsing and converting an HTTP response from an EJBServlet as claimed.

Accordingly, Sharma in view of Tuatini does not teach or suggest parsing and converting the HTTP response by the client library into return information compatible with the non-Java application.

For at least the reasons established above in sections I-IV, Applicants respectfully submit that all of the limitations of independent Claim 1 are not taught or suggested by Sharma in view of Cox and respectfully requests allowance of this claim.

Claims Depending From Claim 1:

Claims 2-10, 12, 13, 15-23, and 31 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini.

Claims 11 and 14 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini in further view of Cox.

Claims 24-30 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini in further view of Haynie.

Dependent Claims 2-31 depend directly or indirectly from independent Claim 1 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established in sections I-IV above, Applicants respectfully submit that all of the limitations of dependent Claims 2-31 are not taught or suggested by Sharma in view of Tuatini and respectfully request allowance of this claim. Applicants respectfully submit that neither Cox or Haynie cure the deficiencies of Sharma in view of Tuatini.

Claim 32:

Claim 32 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini.

Independent Claim 32 remains as originally filed.

Independent claim 32 provides:

32. A computing system for accessing an EJB by a non-Java application comprising:
- a) **a non-Java application** in communication with **a client library**;
 - b) a means for calling the client library from the non-Java application wherein said means for calling the client library is used to establish communication between the non-Java application and the client library;
 - c) an EjbServlet in communication with the client library wherein the client library comprises a function to take input parameter information from the call, embed the information into an HTTP request, and transfer the request to the EjbServlet;
 - d) a means for transferring information between the client library and the EjbServlet wherein said means for transferring it is used to establish communication between the EjbServlet and the client library via an HTTP protocol;
 - e) the EjbServlet configured to receive the HTTP request from the client library and invoke a corresponding method on an EJB; and

- f) a remote method interface (RMI) for invoking methods and returning Java objects between the EjbServlet and the EJB.

Applicants respectfully submit that Claim 32 is not taught or suggested by Sharma in view of Tuatini because they fail to teach or suggest all of the limitations recited Claim 32. Specifically, Sharma in view of Tuatini fails to disclose a non-Java application and a client library.

Independent Claim 32 includes limitations substantially similar to the limitations discussed in sections I, II and IV above. For at least the reasons established above in sections I, II, and IV, Applicants respectfully submit that all of the limitations of independent Claim 32 are not taught or suggested by Sharma in view of Tuatini and respectfully request allowance of this claim.

Claims Depending From Claim 32:

Claims 33-37 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Sharma in view of Tuatini.

Dependent Claims 33-37 depend directly or indirectly from independent Claim 32 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established in sections I, II, and IV above, Applicants respectfully submit that all of the limitations of dependent Claims 33-37 are not taught or suggested by Sharma in view of Tuatini and respectfully request allowance of this claim.

CONCLUSION

Applicants respectfully submit that the present application is in condition for allowance for the reasons stated above. If the Examiner has any questions or comments or otherwise feels it would be helpful in expediting the application, he is encourage to telephone the undersigned at (972) 731-2288.

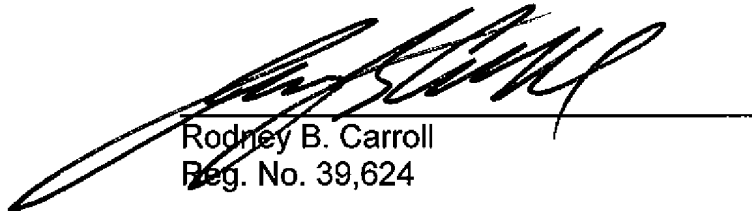
The Commissioner is hereby authorized to charge payment of any further fees associated with any of the foregoing papers submitted herewith, or to credit any overpayment thereof, to Deposit Account No. 21-0765, Sprint.

Respectfully submitted,

Date: _____

1-15-08

CONLEY ROSE, P.C.
5601 Granite Parkway, Suite 750
Plano, Texas 75024
(972) 731-2288
(972) 731-2289 (facsimile)



Rodney B. Carroll
Reg. No. 39,624

ATTORNEY FOR APPLICANTS